REMARKS

Claims 18-20 and 28-30 were pending in the Application. Applicants cancelled claims 18-20 and 28-30 without prejudice or disclaimer and added claims 37-55 which correspond to originally filed claims 18-36. Hence, claims 37-55 are pending.

Applicants' previous amendments, including cancellations, were for the purpose of expediting issuance of a patent. However, since the Examiner has issued a final rejection in the present case and did not allow any claims, Applicants are reinstating originally filed claims 18-36. Applicants will discuss pending claims 37-55 (correspond to originally filed claims 18-36) in connection with the Office Action with a mailing date of May 24, 2007.

In that Office Action, the Examiner objects to claim 38 under 37 C.F.R. §1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. Office Action (5/24/2007), page 2. Applicants respectfully traverse.

Claim 38 clearly does limit the subject matter of claim 37. In claim 38, a computation node in the plurality of interconnected preselected computation nodes, where the plurality of interconnected preselected computation nodes is recited in claim 37, is further defined. Hence, the subject matter of claim 37 is further limited. Further, if the Examiner is asserting that it is improper to have a dependent method claim because it does not specifically introduce a new step or does not specifically state a "step" that is limited, Applicants respectfully traverse. There is no language in 37 C.F.R. §1.75(c) that states that a dependent method claim must introduce a new step or must specifically state a "step" that is limited. Instead, 37 C.F.R. §1.75(c) simply states that a dependent claim further limits another claim in the application, which claim 38 does. Accordingly, Applicants respectfully request the Examiner to withdraw the objections to claim 38.

Further, the Examiner rejects claims 37-39, 43-48 and 50-55 under 35 U.S.C. §102(b) as being anticipated by Requa et al. ("The Piecewise Data Flow Architecture: Architectural Concepts" (hereinafter "Requa"). Applicants respectfully traverse for at least the reasons provided below.

Applicants respectfully assert that Requa does not disclose "assigning a group of instructions selected from the plurality of groups of instructions to a plurality of interconnected preselected computation nodes" as recited in claim 37 and similarly in claim 47. The Examiner cites the second paragraph in the first column on page 426 of Requa as disclosing the above-cited claim limitation. Office Action (5/24/2007), page 3. Applicants respectfully traverse.

Requa instead discloses that instructions are grouped into relatively small blocks (up to 255 instructions per block). Page 426. Requa further disclose that each block includes information on data dependencies within that block. Page 426. Additionally, Requa discloses that when a block executes, each instruction is scheduled for execution on a processor after all its data dependencies have been satisfied. Page 426.

Hence, Requa discloses that when a block executes, each instruction is scheduled for execution on a processor.

There is no language in the cited passage that discloses assigning a group of instructions selected from the plurality of groups of instructions. Instead, Requa discloses scheduling each instruction for execution on a processor when a block executes. Neither is there any language in the cited passage that discloses assigning a group of instructions selected from the plurality of groups of instructions to a plurality of interconnected preselected computation nodes. Thus, Requa does not disclose all of the limitations of claims 37 and 47, and thus Requa does not anticipate claims 37 and 47. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "loading the group of instructions to the plurality of interconnected preselected computation nodes" as recited in claim 37 and similarly in claim 47. The Examiner cites the second paragraph in the first column on page 426 as well as page 433 of Requa as disclosing the above-cited claim limitation. Office Action (5/24/2007), page 3. Applicants respectfully traverse.

As stated above, Requa instead discloses that when a block executes, each instruction is scheduled for execution on a processor. Page 426. Requa further discloses a PDF block processor that supervises the actual execution of blocks that have been

selected for execution. Page 433. Further, Requa discloses that when a block has been selected for execution, the address of that block is sent to the PDF block processor where the block must be read from memory and then undergo a transformation. Page 433. Additionally, Requa discloses that the block processor must allocate a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Page 433. Furthermore, Requa discloses that next, each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed).

Hence, Requa discloses that when a block executes, each instruction is scheduled for execution on a processor. Requa further discloses a PDF block processor that allocates a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Further, Requa discloses that each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed).

There is no language in the cited passages that discloses loading the group of instructions. Neither is there any language in the cited passages that discloses loading the group of instructions to the plurality of interconnected preselected computation nodes. Thus, Requa does not disclose all of the limitations of claims 37 and 47, and thus Requa does not anticipate claims 37 and 47. M.P.E.P. §2131.

Claims 38-39 and 43-46 each recite combinations of features of independent claim 37, and hence claims 38-39 and 43-36 are not anticipated by Requa for at least the above-stated reasons that claim 37 is not anticipated by Requa.

Further, claims 48 and 50-55 each recite combinations of features of independent claim 47, and hence claims 48 and 50-55 are not anticipated by Requa for at least the above-stated reasons that claim 47 is not anticipated by Requa.

For example, Requa does not disclose "wherein at least one computation node included in the plurality of interconnected preselected computation nodes has at least one input port capable of being coupled to at least one preselected first other computation node included in the plurality of interconnected preselected computation nodes, the input

port to receive input data, a first store coupled to the at least one input port to store the input data, a second store coupled to an instruction sequencer, the second store to receive and store the at least one instruction, an instruction wakeup unit to match the input data to the at least one instruction, at least one execution unit to execute the at least one instruction using the input data to produce output data, at least one output port capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes, and a router to direct the output data from the at least one output port to the at least one preselected second other computation node" as recited in claim 38. The Examiner cites the second paragraph in the second column on page 427; page 433; Figure 1 and the PDF block processor as disclosing the above-cited claim limitations. Office Action (5/24/2007), pages 3-4. Applicants respectfully traverse.

Requa instead discloses that the PDF employs an interconnection scheme, called a PDF FIFO queue which connects producers and consumers of information in a synchronous manner, where any consumer can receive any data. Page 427. Requa further discloses a PDF block processor that allocates a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Page 433. Further, Requa discloses that each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed). Page 433.

There is no language in the cited passages or any depiction in Figure 1 that discloses at least one computation node included in a plurality of interconnected preselected computation nodes. Neither is there any language in the cited passages or any depiction in Figure 1 that discloses at least one computation node included in a plurality of interconnected preselected computation nodes has at least one input port capable of being coupled to at least one preselected first other computation node included in the plurality of interconnected preselected computation nodes. Neither is there any language in the cited passages or any depiction in Figure 1 that discloses the input port to receive input data, a first store coupled to the at least one input port to store the input data. Neither is there any language in the cited passages or any depiction in Figure 1 that

discloses a second store coupled to an instruction sequencer. Neither is there any language in the cited passages or any depiction in Figure 1 that discloses that the second store to receive and store the at least one instruction. Neither is there any language in the cited passages or any depiction in Figure 1 that discloses an instruction wakeup unit to match the input data to the at least one instruction. Neither is there any language in the cited passages or any depiction in Figure 1 that discloses at least one output port capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes. Neither is there any language in the cited passages or any depiction in Figure 1 that discloses a router to direct the output data from the at least one output port to the at least one preselected second other computation node. Applicants kindly request the Examiner to particularly point out which elements in Requa allegedly disclose each of these limitations pursuant to 37 C.F.R. §1.104(c)(2).

Applicants further assert that Requa does not disclose "sending at least two instructions selected from the group of instructions from an instruction sequencer to a selected computation node included in the plurality of interconnected preselected computation nodes for storage in a store" as recited in claim 43. The Examiner cites the second paragraph in the first column on page 426 of Requa as disclosing the above-cited claim limitation. Office Action (5/24/2007), page 5. Applicants respectfully traverse.

As stated above, Requa instead discloses that when a block executes, each instruction is scheduled for execution on a processor.

There is no language in the cited passage that discloses sending at least two instructions selected from the group of instructions. Neither is there any language in the cited passage that discloses sending at least two instructions selected from the group of instructions from an instruction sequencer. Neither is there any language in the cited passage that discloses sending at least two instructions selected from the group of instructions from an instruction sequencer to a selected computation node included in the plurality of interconnected preselected computation nodes for storage in a store. Thus,

Requa does not disclose all of the limitations of claim 43, and thus Requa does not anticipate claim 43. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node included in the plurality of interconnected preselected computation nodes" as recited in claim 44. The Examiner cites the second paragraph in the second column on page 427 as disclosing the above-cited claim limitation. Office Action (5/24/2007), page 5. Applicants respectfully traverse.

Requa instead discloses that the PDF employs an interconnection scheme, called a PDF FIFO queue which connects producers and consumers of information in a synchronous manner, where any consumer can receive any data. Page 427. Requa further discloses that instructions waiting for execution reside in special registers waiting for all data dependencies to be satisfied. Page 427.

Hence, Requa discloses that the Piecewise Data Flow Architecture (PDF) employs an interconnection scheme which connects producers and consumers of information. Further, Requa discloses that instructions waiting for execution reside in special registers waiting for all data dependencies to be satisfied.

There is no language in the cited passage that discloses the aspect of matching. Neither is there any language in the cited passage that discloses matching at least one instruction selected from the group of instructions. Neither is there any language in the cited passage that discloses matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node. Neither is there any language in the cited passage that discloses matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node included in the plurality of interconnected preselected computation nodes. Thus, Requa does not disclose all of the limitations of claim 44, and thus Requa does not anticipate claim 44. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "sending a first set of instructions selected from a first group of instructions selected from the plurality of

groups of instructions from an instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a first frame included in a first computation node included in the plurality of interconnected preselected computation nodes; and sending a second set of instructions selected from the first group of instructions from the instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a second frame included in the first computation node" as recited in claim 45. The Examiner cites paragraph 1 on page 433 of Requa as disclosing the above-cited claim limitations. Office Action (5/24/2007), page 6. Applicants respectfully traverse.

As stated above, Requa instead discloses a PDF block processor that supervises the actual execution of blocks that have been selected for execution. Page 433. Further, Requa discloses that when a block has been selected for execution, the address of that block is sent to the PDF block processor where the block must be read from memory and then undergo a transformation. Page 433. Additionally, Requa discloses that the block processor must allocate a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Page 433. Furthermore, Requa discloses that next, each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed).

Hence, Requa discloses a PDF block processor that allocates a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Further, Requa discloses that each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed).

There is no language in the cited passage that discloses sending a first set of instructions selected from a first group of instructions. Neither is there any language in the cited passage that discloses sending a first set of instructions selected from a first group of instructions selected from the plurality of groups of instructions. Neither is there any language in the cited passage that discloses sending a first set of instructions selected from a first group of instructions selected from the plurality of groups of

instructions from an instruction sequencer. Neither is there any language in the cited passage that discloses sending a first set of instructions selected from a first group of instructions selected from the plurality of groups of instructions from an instruction sequencer to the plurality of interconnected preselected computation nodes for storage. Neither is there any language in the cited passage that discloses sending a first set of instructions selected from a first group of instructions selected from the plurality of groups of instructions from an instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a first frame. Neither is there any language in the cited passage that discloses sending a first set of instructions selected from a first group of instructions selected from the plurality of groups of instructions from an instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a first frame included in a first computation node included in the plurality of interconnected preselected computation nodes. Neither is there any language in the cited passage that discloses sending a second set of instructions selected from the first group of instructions. Neither is there any language in the cited passage that discloses sending a second set of instructions selected from the first group of instructions from the instruction sequencer. Neither is there any language in the cited passage that discloses sending a second set of instructions selected from the first group of instructions from the instruction sequencer to the plurality of interconnected preselected computation nodes for storage. Neither is there any language in the cited passage that discloses sending a second set of instructions selected from the first group of instructions from the instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a second frame. Neither is there any language in the cited passage that discloses sending a second set of instructions selected from the first group of instructions from the instruction sequencer to the plurality of interconnected preselected computation nodes for storage in a second frame included in the first computation node.

Thus, Requa does not disclose all of the limitations of claim 45, and thus Requa does not anticipate claim 45. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "assigning a first group of instructions to a first set of frames included in the plurality of interconnected preselected

computation nodes; assigning a second group of instructions to a second set of frames included in the plurality of interconnected preselected computation nodes, wherein the first group and the second group of instructions are capable of concurrent execution, and wherein at least one output datum associated with the first group of instructions is written to a register file and passed directly to the second group of instructions for use as an input datum by the second group of instructions" as recited in claim 46. The Examiner cites the second paragraph in the first column on page 430; and paragraph 1 on page 433 of Requa as disclosing the above-cited claim limitations. Office Action (5/24/2007), pages 6-7. Applicants respectfully traverse.

Requa instead discloses that once an instruction completes, any instructions that are instruction-dependent on it must be notified and any instructions data dependent on it must read the result before the instruction/result register may be released. Page 430. Requa further discloses a PDF block processor that supervises the actual execution of blocks that have been selected for execution. Page 433. Further, Requa discloses that when a block has been selected for execution, the address of that block is sent to the PDF block processor where the block must be read from memory and then undergo a transformation. Page 433. Additionally, Requa discloses that the block processor must allocate a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Page 433. Furthermore, Requa discloses that next, each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed).

Hence, Requa discloses that once an instruction completes, any instructions that are instruction-dependent on it must be notified and any instructions data dependent on it must read the result before the instruction/result register may be released. Requa additionally discloses a PDF block processor that allocates a different register for each instruction in the block and then modify the operand source fields of each instruction to reflect the registers that will contain the needed values. Further, Requa discloses that each instruction is sent to the instruction issue section, to await the availability of its input operands (after which it is executed).

There is no language in the cited passages that discloses assigning a first group of instructions to a first set of frames. Neither is there any language it the cited passages that discloses assigning a first group of instructions to a first set of frames included in the plurality of interconnected preselected computation nodes. Neither is there any language it the cited passages that discloses assigning a second group of instructions to a second set of frames. Neither is there any language it the cited passages that discloses assigning a second group of instructions to a second set of frames included in the plurality of interconnected preselected computation nodes. Neither is there any language it the cited passages that discloses that the first group and the second group of instructions are capable of concurrent execution. Neither is there any language it the cited passages that discloses that at least one output datum associated with the first group of instructions is written to a register file and passed directly to the second group of instructions for use as an input datum by the second group of instruction.

Thus, Requa does not disclose all of the limitations of claim 46, and thus Requa does not anticipate claim 46. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "statically assigning all of the plurality of groups of instructions for execution" as recited in claim 50. The Examiner cites Figure 8 and column 1, paragraph 2 on page 432 of Requa as disclosing the above-cited claim limitation. Office Action (5/24/2007), page 8. Applicants respectfully traverse.

Requa instead discloses that analysis of possible block overlaps by a language processor is difficult because only a static view of the program is available. Page 432.

There is no language in the cited passage that discloses statically assigning all of the plurality of groups of instructions for execution. The Examiner appears to have focused on the fact that the term "static" is used. However, the Examiner has not provided any rational basis for concluding that a static view of a program, as taught in Requa, is the same as statically assigning all of the plurality of groups of instructions for execution. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that a static

view of a program, as taught in Requa, is the same as statically assigning all of the plurality of groups of instructions for execution, and that it would be so recognized by persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Since the Examiner has not provided any such objective evidence, the Examiner has not presented a *prima facie* case of anticipation in rejecting claim 50. M.P.E.P. §2112.

Applicants further assert that Requa does not disclose "dynamically issuing one or more instructions from at least one of the plurality of groups of instructions for execution" as recited in claim 51. The Examiner cites the second column, third paragraph on page 426 of Requa as disclosing the above-cited claim limitation. Office Action (5/24/2007), page 8. Applicants respectfully traverse.

Requa instead discloses that to get data flow-like execution, the data dependencies within a basic block are encoded as part of the instructions. Page 426. Requa further discloses that instructions are sent to the hardware scheduler only after all data dependencies have been satisfied. Page 426.

There is no language in the cited passage that discloses dynamically issuing one or more instructions from at least one of the plurality of groups of instructions for execution. Thus, Requa does not disclose all of the limitations of claim 51, and thus Requa does not anticipate claim 51. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "generating a wakeup token to reserve an output data channel to connect selected computation nodes included in the plurality of interconnected preselected computation nodes" as recited in claim 52. The Examiner cites the second paragraph of the second column on page 427 of Requa as disclosing above-cited claim limitation. Office Action (5/24/2007), page 9. Applicants respectfully traverse.

As stated above, Requa instead discloses that the PDF employs an interconnection scheme, called a PDF FIFO queue which connects producers and consumers of information in a synchronous manner, where any consumer can receive any data. Page 427. Requa further discloses that instructions waiting for execution reside in special registers waiting for all data dependencies to be satisfied. Page 427.

Hence, Requa discloses that the Piecewise Data Flow Architecture (PDF) employs an interconnection scheme which connects producers and consumers of information. Further, Requa discloses that instructions waiting for execution reside in special registers waiting for all data dependencies to be satisfied.

There is no language in the cited passage that discloses generating a wakeup token. Neither is there any language in the cited passage that discloses generating a wakeup token to reserve an output data channel to connect selected computation nodes. Neither is there any language in the cited passage that discloses generating a wakeup token to reserve an output data channel to connect selected computation nodes included in the plurality of interconnected preselected computation nodes. Thus, Requa does not disclose all of the limitations of claim 52, and thus Requa does not anticipate claim 52. M.P.E.P. §2131.

Applicants further assert that Requa does not disclose "routing an output datum arising from executing the group of instructions to a consumer node included in the plurality of interconnected preselected computation nodes, wherein the address of the consumer node is included in a token associated with at least one instruction included in the group of instructions" as recited in claim 55. The Examiner cites the second paragraph in the second column on page 429 of Requa as disclosing the above-cited claim limitations. Office Action (5/24/2007), page 10. Applicants respectfully traverse.

Requa instead discloses that the operation code is used for two purposes, namely, it routes the instruction to a processor of the required generic type and specifies the operation the processor is to perform. Page 429.

There is no language in the cited passage that discloses routing an output datum arising from executing the group of instructions to a consumer node included in the plurality of interconnected preselected computation nodes. Instead, Requa discloses an operation code that is used for routing an instruction to a processor. Thus, Requa does not disclose all of the limitations of claim 54, and thus Requa does not anticipate claim 54. M.P.E.P. §2131.

As a result of the foregoing, Applicants respectfully assert that not each and every claim limitation was found within Requa, and thus claims 37-39, 43-48 and 50-55 are not anticipated by Requa. M.P.E.P. §2131.

Further, the Examiner rejects claims 40 and 41 under 35 U.S.C. §103(a) as being unpatentable over Requa in view of Official Notice. Office Action (5/24/20070, page 10. Applicants respectfully traverse for at least the reasons stated below.

The Examiner admits that Requa does not teach the limitations of claims 40 ("wherein at least one of the plurality of groups of instructions is a hyperblock") and 41 ("wherein at least one of the plurality of groups of instructions is a superblock"). However, the Examiner asserts that one of ordinary skill in the art would be capable and motivated to use hyperblocks or superblocks in replace of basic blocks. Office Action (5/24/2007), page 11. Applicants respectfully traverse.

In order to establish a *prima facie* case of obviousness, the Examiner must show reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed. *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1458 (Fed. Cir. 1998). The Examiner must provide articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (cited approvingly in *KSR International Co. v. Teleflex Inc.*, 82 U.S.P.Q.2d 1385, 1396 (U.S. 2007)).

The Examiner asserts that one of ordinary skill in the art would be capable of replacing the basic block with either a hyperblock or a superblock to take advantage of the ability to have multiple exits from a block. Office Action (5/24/2007), page 11. However, Requa teaches a basic block is a standard compiler term for a sequence of instructions that has no branching into or out of the block except at the beginning and/or ending. Page 426. Requa further teaches that this definition is significant because once the first instruction in the block begins execution, we know that all remaining instructions in the block must be executed soon. *Id.* Hence, Requa specifically teaches against having multiple exits from a block. Thus, the Examiner's rationale does not

provide reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would modify Requa to include the above-cited missing claim limitations of claims 40 and 41. Accordingly, the Examiner has not presented a *prima facie* case of obviousness for rejecting claims 40 and 41. *KSR International Co. v. Teleflex Inc.*, 82 U.S.P.Q.2d 1385, 1396 (U.S. 2007).

Additionally, the Examiner rejects claims 42 and 49 under 35 U.S.C. §103(a) as being unpatentable over Requa in view of Fisher ("Trace Scheduling: A Technique for Global Microcode Compaction").

Claim 42 depends from claim 37, and hence claim 42 is allowable over Requa in view of Fisher for at least the above-stated reasons that claim 37 is not anticipated by Requa.

Further, claim 49 depends from claim 47, and hence claim 49 is allowable over Requa in view of Fisher for at least the above-stated reasons that claim 47 is not anticipated by Requa.

CONCLUSION

As a result of the foregoing, it is asserted by Applicants that claims 37-55 in the Application are in condition for allowance, and respectfully request an allowance of such claims. Applicants respectfully request that the Examiner call Applicants' attorney at the below listed number if the Examiner believes that such a discussion would be helpful in resolving any remaining issues.

Respectfully submitted,

WINSTEAD P.C.

Attorneys for Applicants

By: _____
Robert A. Voigt, Jr.
Reg. No. 47,159

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

Austin_1\534200\1
22084-P033US 5/26/2008